# Lesson 1

Big Data I.S.P.M.

<https://azure.microsoft.com/en-us/solutions/big-data/>

Big data: Massive amount of data which cannot be stored, processed and analyzed using the traditional ways.

Big data 5 V’s: **Volume, Veracity, Value, Velocity, Variety**

Advanced analytics on big data:

622

522

422

322

222

1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tool** | | **Ingest** | **Store** | **Prep & Train** | **Model & Serve** |
| Icon  Description automatically generated | **Azure Data Factory** | ✓ |  |  |  |
|  | **Azure Synapse Analytics**  **(Azure SQL DW)** | ✓ |  |  | ✓ |
| Logo, icon  Description automatically generated | **Azure Stream Analytics** | ✓ |  | ✓ |  |
| Icon  Description automatically generated | **Azure Files** |  | ✓ |  |  |
| Icon  Description automatically generated | **Azure Queue** |  | ✓ |  |  |
| Icon  Description automatically generated | **Azure Data Lake Storage Gen 2**  (Non-relational data store) |  | ✓ |  |  |
| **A picture containing icon  Description automatically generated** | **CosmosDB**  (Non-relational data store) |  | ✓ |  | ✓ |
| Icon  Description automatically generated | **Azure Blob (WASB)**  (Non-relational data store) |  | ✓ |  |  |
| Logo  Description automatically generated | **Azure HDInsight**  **(Hadoop for azure)** |  |  | ✓ |  |
| **Icon  Description automatically generated** | **Azure Data Lake Analytics** |  |  | ✓ |  |
| **Icon  Description automatically generated** | **Azure Databricks**  **(python, Scala, Spark SQL, Spark R, Spark Structured Streaming)** |  |  | ✓ |  |
| **Icon  Description automatically generated** | **Power BI** |  |  |  | ✓ |

# Lesson 2

Data types & Storage

**Structured data**

- Structure is defined at design time.

- Data structure is designed in the form of tables.

**Semi-structured data.**

- Non-relational or NoSQL data (won’t fit neatly into tables, rows, and columns) – Uses tags or keys that organize and provide a hierarchy for the data

**Nonstructured data**

- Examples of nonstructured data include binary, audio, and image files

- The data structure is defined only when the data is read.

- Nonrelational systems can also support semistructured data such as JSON file formats.

- The schema of unstructured data is typically defined at query time. This means that data can be loaded onto a data platform in its native format.

Implement Relational Data Stores

The open-source world offers four types of NoSQL databases:

1. Key-value store: Stores key-value pairs of data in a table structure.
2. Document database: Stores documents that are tagged with metadata to aid document searches.
3. Graph database: Finds relationships between data points by using a structure that's composed of vertices and edges.
4. Column database: Stores data based on columns rather than rows. Columns can be defined at the query's runtime, allowing flexibility in the data that's returned performantly.

# Lesson 3

Data Storage

<https://docs.microsoft.com/en-us/learn/modules/intro-to-data-in-azure/3-how-azure-storage-meets-your-business-storage-needs>

[Azure] Four Storage options:

**Azure Blob (WASB):** A scalable object store for text and binary data

This is a data store that will store but not query data, your cheapest option is to set up a storage account as a Blob store.

Blob storage works well with images and unstructured data

Flexible pricing options (cold vs hot storage)

Icon

Description automatically generated

Icon

Description automatically generated

**Azure Files:** Managed file shares for cloud or on-premises deployments. Accessible via the industry standard Server Message Block (SMB) protocol

Icon

Description automatically generated

**Azure Queue:** Azure Queue storage is a service for storing large numbers of messages that can be accessed from anywhere in the world.

Azure Table: A NoSQL store for no-schema storage of structured data

Diagram, icon

Description automatically generated

Azure Storage Tables is aimed at high capacity on a single region (optional secondary read only region but no failover), indexing by PK/RK and storage optimized pricing;

Azure Cosmos DB is a globally distributed database service.

**high throughput** (single-digit millisecond latency),

**global distribution** (multiple failover), SLA-backed predictive performance with automatic indexing of each attribute/property and a pricing model focused on throughput.

**A picture containing icon

Description automatically generated**

Implement non-relational data stores:

- Azure Data Lake v2

- WASB (Blob storage)

- CosmosDB

# Lesson 4

SQL

Azure SQL

Database elastic pools are a simple, cost-effective solution for managing and scaling multiple databases that have varying and unpredictable usage demands. The databases in an elastic pool are on a single server and share a set number of resources at a set price.

SQL on-premises

# Lesson 5

Storage Redundancy

[Azure] Storage Redundancy

|  |  |  |
| --- | --- | --- |
| **Abbr** | **Name** | **Description** |
| LRS | Locally Redundant Storage | Replicates your data three times within a single data center |
| ZRS | Zone-Redundant Storage | Replicates your data across three storage clusters in a single region. |
| GRS | Geo-Redundant Storage | Replicates your data to a secondary region. Can withstand regional outage. |
| RA-GRS | Read-Access Geo Redundant Storage | Provides read-only access to the data in the secondary location, in addition to GRS. |
| GZRS | Geo-Zone-Redundant Storage | Replicates data across three Azure Availability Zones in two regions. |
| RA-GZRS | Read-Access Geo Zone Redundant Storage | Provides read-only access to the data in the secondary location, in addition to GZRS. |

[Azure] Storage Redundancy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Abbr** | LRS | ZRS | GRS | GZRS |
| Node Unavailability (within a DC) | Yes | Yes | Yes | Yes |
| DC outage |  | Yes | Yes | Yes |
| Region Outage |  |  | Yes | Yes |
| Read-Access in event of a Region Outage |  |  | RA-GRS | RA-GZRS |
| Durability over a given year | 11 9’s | 12 9’s | 16 9’s | 16 9’s |

# Lesson 6

Data Platforms

[Azure] Azure Cosmos DB is a globally distributed, multi-model database. You can deploy it by using several API models:

SQL API MongoDB API Cassandra API Gremlin API Table API

Because of the multi-model architecture of Azure Cosmos DB, you benefit from each model's inherent capabilities. For example, you can use MongoDB for semi-structured data, Cassandra for wide columns, or Gremlin for graph databases. Because of the multi-model architecture of Azure Cosmos DB, you benefit from each model's inherent capabilities. For example, you can use MongoDB for semi-structured data, Cassandra for wide columns, or Gremlin for graph databases.

[Azure] Azure SQL Database

**Use T-SQL to query the contents of a SQL Database.**

*Remember this is Azure SQL Database PaaS not an instance installed within Azure virtual machine*

Use SQL Database when you need to scale up and scale down OLTP systems on demand. SQL Database is a good solution when your organization wants to take advantage of Azure security and availability features

Data Services

<https://docs.microsoft.com/en-us/learn/modules/survey-the-azure-data-platform/10-azure-other-data-platform-services>

Logo, company name

Description automatically generated

[Azure] Databricks

Serverless platform that's optimized for Azure. One-click setup

Streamlined workflows

Interactive workspace + fully managed Spark clusters for Spark-based applications.

In Databricks notebooks you'll use familiar programming tools such as R, Python, Scala, and SQL

Icon

Description automatically generated

[Azure] Data Factory

**Organize raw data into meaningful data stores and data lakes**

Orchestrates the movement of data between various data stores.

Cloud-integration service.

Streamlined workflows

processes and transforms data by using compute services such as Azure HDInsight, Hadoop, Spark, and Azure Machine Learning

Graphical user interface, application, Teams

Description automatically generated

[Azure] Data Catalog

**Discover, understand, and consume data sources**

Is a fully managed cloud service

Is the best choice to store documentation about a data source

# Lesson 7

The Data Engineering Process

<https://docs.microsoft.com/en-us/learn/modules/data-engineering-processes/3-data-engineering-practices>

**ETL**

[Azure] Azure Data Factory v2



**ELT**

[Azure] Azure Data Factory v2

[Azure] Azure Synapse

[Azure] HDInsight with Hive

[Azure] Oozie on HDInsight

SQL Server Integration Services (SSIS)



**Extract**

During the extraction process, data engineers define the data and its source:

[1] Define the data source: Identify source details such as the resource group, subscription, and identity information such as a key or secret.

[2] Define the data: Identify the data to be extracted. Define data by using a database query, a set of files, or an Azure Blob storage name for blob storage.

**Transform**

[3] Define the data transformation: Data transformation operations can include splitting, combining, deriving, adding, removing, or pivoting columns. Map fields between the data source and the data destination. You might also need to aggregate or merge data.

**Load**

[4] Define the destination: During a load, many Azure destinations can accept data formatted as a JavaScript Object Notation (JSON), file, or blob. You might need to write code to interact with application APIs.

Azure Data Factory offers built-in support for Azure Functions. You'll also find support for many programming languages, including Node.js, .NET, Python, and Java. Although Extensible Markup Language (XML) was common in the past, most systems have migrated to JSON because of its flexibility as a semistructured data type.

[5] Start the job: Test the ETL job in a development or test environment. Then migrate the job to a production environment to load the production system.

[6] Monitor the job: ETL operations can involve many complex processes. Set up a proactive and reactive monitoring system to provide information when things go wrong. Set up logging according to the technology that will use it.

# Lesson 8

INGESTION: Data Ingestion

<https://docs.microsoft.com/en-us/learn/modules/explore-data-ingestion-azure/2-describe-common-practices-for-data-loading>

Data ingestion is the first part of any data warehousing solution. It is arguably the most important part. In a big data system, data ingestion has to be fast enough to capture the large quantities data that may be heading your way, and have enough compute power to process this data in a timely manner.

Ways to ingest data:

**Azure Data Factory v2** [Azure] **ELT + ETL**



**PolyBase** [Azure]

*Azure SQL Database does not support PolyBase*

**SQL Server Integration Services ELT + ETL**

Logo

Description automatically generated

**Icon

Description automatically generated**

**Azure Databricks** [Azure]



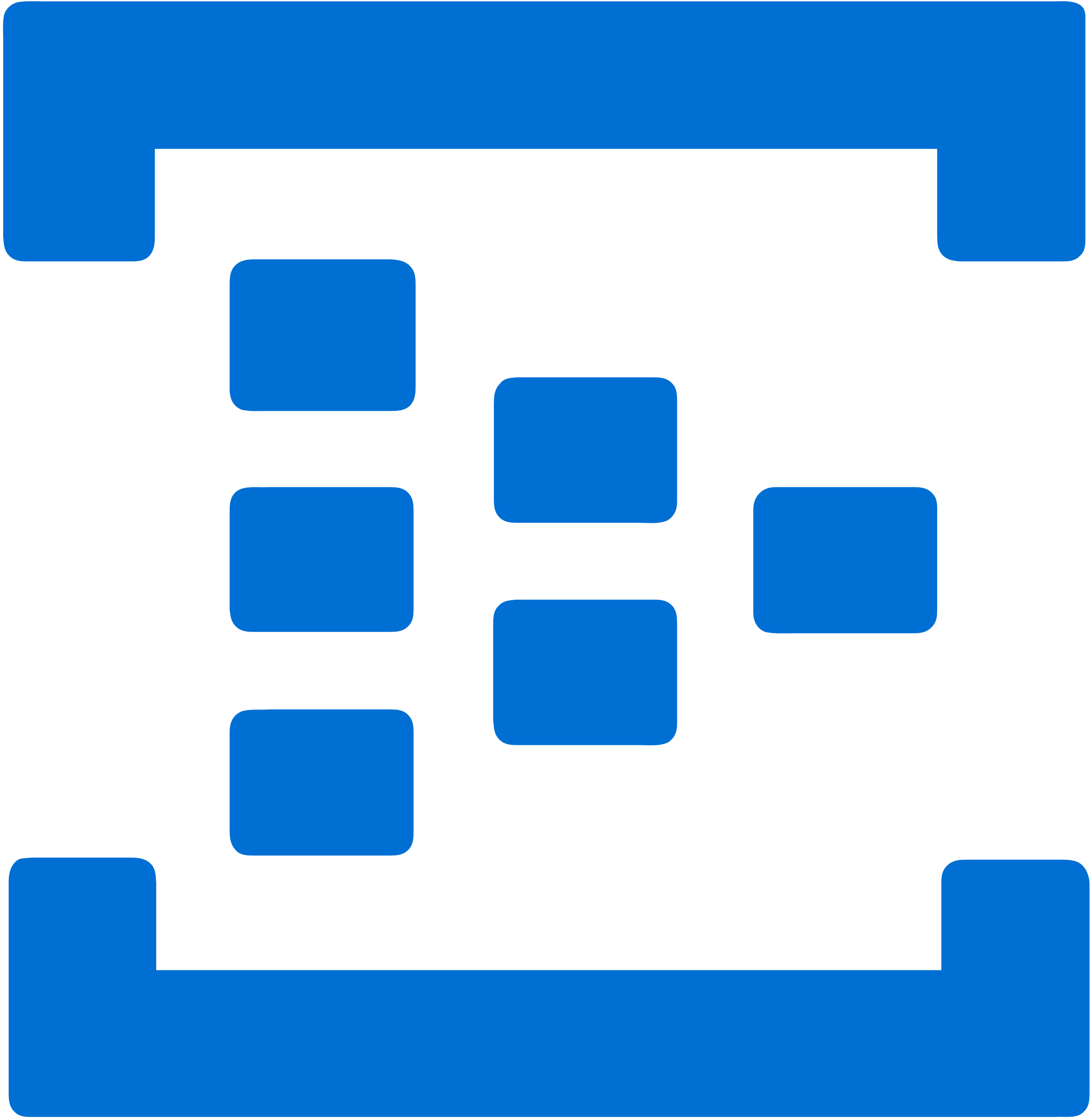
**Azure Synapse Analytics** [Azure]

Icon

Description automatically generated

**IoT Hub** [Azure]

Communication: device -> cloud & cloud -> device



**Event Hub** [Azure]

# Lesson 9

INGESTION: Azure Data Factory

Icon

Description automatically generated**Azure Data Factory v2** [Azure] **ELT + ETL**

Data Factory contains a series of interconnected systems that provide a complete end-to-end platform for data engineers.

As it ingests the data, Data Factory can clean, transform, and restructure the data, before loading it into a repository such as a data warehouse. Once the data is in the data warehouse, you can analyze it.

Data Factory provides an orchestration engine. Orchestration is the process of directing and controlling other services, and connecting them together, to allow data to flow between them

Diagram

Description automatically generated

**PolyBase** [Azure]

*Azure SQL Database does not support PolyBase*

Feature of SQL Server and Azure Synapse Analytics

Enables you to run Transact-SQL queries that read data from external data sources (makes these sources appear like SQL tables).

Data Factory can directly invoke PolyBase on your behalf if your data is in a PolyBase-compatible data store.

Logo

Description automatically generated**SQL Server Integration Services ELT + ETL**

*SSIS is an on-premises utility.*

use SSIS to solve complex business problems by copying or downloading files, loading data warehouses, cleaning and mining data, and managing SQL database objects and data. SSIS is part of Microsoft SQL Server.

How to ingest data from Azure SQL database into ADLS gen 2 using Data Factory:

2

Pre-requisite: ADLS gen 2

(should have the following configured)

Storage account + + +

Images adlsfactory adlsserver

A picture containing icon

Description automatically generatedIcon

Description automatically generatedIcon

Description automatically generated

NB: remember to toggle firewall rules in [Azure] SQL server

“Allow Azure services and resources to access this server”

Graphical user interface, text, application, chat or text message

Description automatically generated

A picture containing icon

Description automatically generated

[1] Add new linked services:

Manage > Linked Services > New> Azure SQL Database

-- test connection

Manage > Linked Services > New> Azure Data Lake Storage Gen2

-- test connection

Icon

Description automatically generated

Images

Author and Monitor Data Factory [Azure]

*NB: This can have issues. Try using incognito mode in your browser*

[2] Add datasets

*You should have a table already created in your db*

-- Add new dataset

Author > Azure SQL Database

-- Add new dataset

Author > Azure Data Lake Storage Gen2

Icon

Description automatically generated

[3] Create data Pipeline

-- Create “copy data” activity

-- Map between source and destination

-- Trigger pipeline

Exam Prep:

Azure Data Factory

1. Understand the difference between all the available Integration runtime. Pay special attention to the self-hosted integration runtime.
2. Azure Data Factory Copy Activity: Find out schema mapping ways between source & sink. hands-on

# Lesson 10

INGESTION: Azure Synapse Analytics

**Azure Synapse Analytics** [Azure]

Azure Synapse Analytics is generalized analytics service. You can use it to read data from many sources, process this data, generate various analyses and models, and save the results. Azure Synapse Analytics uses a clustered architecture.

You can select between two technologies to process data:

* Transact-SQL
* Spark (same open-source technology used to power databricks)

**Synapse as a massively parallel processing database**

How to load data into Synapse Analytics: houseprices.csv

<https://docs.microsoft.com/en-us/learn/modules/explore-data-ingestion-azure/3-load-data>



1] Create Storage Account

*We use a storage account because ADLS is built on top of blob storage WASB*

storagedata

[2] Add File Share

housingdata

-- Upload csv file

houseprices.csv

[3] Create Synapse workspace Account

synapsews

-- Create Storage Account

synapsews-store

-- Add File Share

synapsews-data

[+] Create new SQL pool

synapsews-pool

Launch Synapse Studio

[+] Create new table

Housingdata

-- Publish table

Create new linked service (Source)

Create “Copy data” pipeline

(Destination)

[2] Create new Data Factory

synapsews-datafactory



*\*File share is like a replacement for a file server*

PREP & TRAIN: Azure Synapse Analytics

<https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/cheat-sheet>

Best Practice for building Azure Synapse Analytics Solutions **M.D.I.P.I**:

Diagram

Description automatically generated

|  |  |
| --- | --- |
| **Design** | **Recommendation** |
| Distribution | Round Robin |
| Indexing | Heap |
| Partition | None |
| Resource Class | largerc or xlargerc |

**Data migration**

First, load your data into Azure Data Lake Storage or Azure Blob Storage. Next, use the COPY statement (preview) to load your data into staging tables. Use the following configuration:

**Distributed or replicated tables**

Start with Round Robin, but aspire to a hash distribution strategy to take advantage of a massively parallel architecture. A distributed table appears as a single table, but the rows are stored across 60 distributions. The rows are distributed with a hash or round-robin algorithm.

|  |  |  |
| --- | --- | --- |
| **Type** | **Great fit for…** | **Watch out if** |
| Replicated | \* Small dimension tables in star schema (<2GB) | \* Many write transactions (insert, upsert, delete, update)  \* You change warehouse units (DWU) and provision frequently |
| Round Robin | \* Temp/Staging Table  No obvious joining key | \* Performance is slow due to data movement |
| Hash | \* Fact Tables  \* Large Dimension Tables | \* The distribution key cannot be updated |

**Index your tables**

Indexing is helpful for reading tables quickly. There is a unique set of technologies that you can use based on your needs:

|  |  |  |
| --- | --- | --- |
| **Type** | **Great fit for…** | **Watch out if** |
| Heap | Temp/Staging Table | \* Any lookup scans the full table |
| Clustered Index | Tables with up to 100 million rows  Large tables (more than 100 million rows) with only 1-2 columns heavily used | \* Used on a replicated table \* You have complex queries involving multiple join and Group By operations \* You make updates on the indexed columns: it takes memory |
| Clustered columnstore index (CGI) (default) | Large tables (more than 100 million rows) | \* Used on a replicated table  \* You make massive update operations on your table |

**Partitioning**

You might partition your table when you have a large fact table (greater than 1 billion rows). In 99 percent of cases, the partition key should be based on date. Be careful to not overpartition, especially when you have a clustered columnstore index.

With staging tables that require ELT, you can benefit from partitioning. It facilitates data lifecycle management. Be careful not to overpartition your data, especially on a clustered columnstore index.

**Incremental load**

If you're going to incrementally load your data, first make sure that you allocate larger resource classes to loading your data. This is particularly important when loading into tables with clustered columnstore indexes. See resource classes for further details.

We recommend using PolyBase and ADF V2 for automating your ELT pipelines into your data warehouse.

Exam prep:

Azure SQL Server & SQL Server Data Warehouse (Synapse Analytics)

1. Which SQL option should I choose?
2. Export an Azure SQL database to a BACPAC file. hands-on
3. Learn about how to secure sensitive data in a SQL database with database encryption by using the Always Encrypted wizard. hands-on
4. Experience Azure SQL Database Advanced Threat Protection features & steps to enable it. hands-on
5. Experiment and enable TDE (Transparent Data Encryption) and keep a note on the steps: Formula(memory trick): MCED — Master Key, Certificate, Encryption & Apply encryption on the DB. hands-on
6. Do an experiment using Powershell & Azure cloud shell. hands-on
7. IP firewall rules. hands-on
8. Read about dynamic data masking for Azure SQL Database and Azure Synapse Analytics. Give special attention to the in-built masking functions & their appropriate usages (Default, Credit Card, Email, Random Number, Custom Text)
9. Polybase: Please execute this hands-on experiment multiple times to load the data from ADLS into WH and memorize all the steps in the correct sequence. Formula(Memory trick): MCSFTL — Master, Credential, Source, File, Table, Load(CTAS). Load New York Taxicab dataset hands-on
10. DW performance benchmarking: This example demonstrates DW performance benchmarking and concluded to utilize a methodology of CTAS and partition switching in lieu of UPDATE and DELETE operations wherever possible. Get a full understanding of this fundamental approach. hands-on

# Lesson 11

INGESTION: Azure Databricks

**Azure Databricks** [Azure]



Azure Databricks is an analytics platform optimized for the Microsoft Azure cloud services platform. Databricks is based on Spark, and is integrated with Azure to streamline workflows.

It provides an interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.

Databricks can process data held in many different types of storage, including Azure Blob storage, Azure Data Lake Store, Hadoop storage, flat files, SQL databases, and data warehouses, and Azure services such as Cosmos DB. Databricks can also process streaming data

**Cluster configurations**

|  |  |
| --- | --- |
| **Standard** | **High Concurrency** |
| Single User | Multiple Users |
| SQL, Python and Scala | SQL, Python and R |
|  | Notebook Isolation |

Exam prep:

Azure Databricks

1. Learn about the technology choices for batch processing and what is the decision criteria to choose one over the others.
2. ETL using Azure Databricks. Special attention to “Load data into Azure SQL Data Warehouse” hands-on
3. Experiment on different cluster configurations. hands-on

# Lesson 12

Hadoop

A picture containing drawing

Description automatically generatedHadoop was developed to solve the following:

|  |  |
| --- | --- |
| **Challenges** | **Solutions** |
| Single central storage | Distributed storage |
| Serial (linear) processing | Map reduce: Parallel processing |
| Lack of ability to process unstructured data | Ability to process every type of data |

Hadoop is an open-source framework that manages big data storage in a distributed way and processes it parallelly. It does this processing and analysis of big data on clusters. The Hadoop technology stack includes related software and utilities including Apache Hive, Apache HBase, Spark, Kafka etc.

Apache Hadoop framework:

**Hadoop Common**

**Hadoop Distributed File System (HDFS)**

**Hadoop YARN**

**Hadoop MapReduce**

Diagram

Description automatically generated

A picture containing drawing

Description automatically generated

Storage unit

HDFS : Specially designed for storing huge datasets in commodity hardware

Master/ name node

-- slave/ data node

-- slave/ data node

-- slave/ data node

Processing unit

Map Reduce: is a programming technique where huge data is processed in a parallel and distributed fashion (data is processed at slave nodes)

Master/ name node

-- slave/ data node

-- slave/ data node

-- slave/ data node

How to install Hadoop (Pseudo-Distributed Mode)

*Practically, you would use something like docker to set this up*

<https://medium.com/@thedsa.in/install-hadoop-3-2-setting-up-a-single-node-hadoop-cluster-22a5754bd9fc>

[1] Create CentOS VM

-- Create Hadoop user

adduser hduser

passwd hduser

usermod -aG wheel hduser

[2] Install Java (CentOS)

<https://www.liquidweb.com/kb/install-java-8-on-centos-7/>

yum -y update

yum install java-1.8.0-openjdk

yum install java-1.8.0-openjdk-headless

[3] Setup SSH

Install OpenSSH Server

sudo yum –y install openssh-server openssh-clients

# start SSH daemon on the openSSH server

sudo systemctl start sshd

sudo systemctl status sshd

[3] Configure SSH key-based Authentication

su – hduser

ssh-keygen -t rsa

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

chmod 640 ~/.ssh/authorized\_keys

# test user

ssh localhost

[4] Download and Configure Hadoop

wget http://apachemirror.wuchna.com/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz

tar -xvzf hadoop-3.2.1.tar.gz

mv hadoop-3.2.1 hadoop

[5] Configure .bashrc and env variables

update-alternatives --config java

# copy the above path

vim .bashrc

#paste above path

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.191.b12-1.el7\_6.x86\_64/jre/

export HADOOP\_HOME=/home/hduser/hadoop

export HADOOP\_INSTALL=$HADOOP\_HOME

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export HADOOP\_YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin

export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib/native"

source ~/.bashrc

echo $JAVA\_HOME

[6] Configure Hadoop

mkdir -p ~/hadoopdata/hdfs/namenode

mkdir -p ~/hadoopdata/hdfs/datanode

vim $HADOOP\_HOME/etc/hadoop/core-site.xml

vim $HADOOP\_HOME/etc/hadoop/hdfs-site.xml

vim $HADOOP\_HOME/etc/hadoop/mapred-site.xml

vim $HADOOP\_HOME/etc/hadoop/yarn-site.xml

[7] Start Hadoop Cluster

hdfs namenode -format

# the above will give you the hostname (VM name)

# SHUTDOWN\_MSG: Shutting down NameNode at AN-01/45.58.38.202

start-dfs.sh

start-yarn.sh

# run the jps command. You should see 6 services running

jps

[7] Configure Firewall

# OPTIONAL: Enable Firewalld

systemctl enable firewalld

systemctl start firewalld

systemctl status firewalld

# allow connections through firewall

sudo su

firewall-cmd --permanent --add-port=9870/tcp

firewall-cmd --permanent --add-port=8088/tcp

firewall-cmd --reload

Forward ports (~/.ssh/config)

Host AN\*

User hduser

Compression yes

ForwardAgent yes

Host AN-01

HostName 102.37.120.33

# NameNode

LocalForward 0.0.0.0:9870 localhost:9870

# ResourceManager

LocalForward 0.0.0.0:8088 localhost:8088

# NodeManager

LocalForward 0.0.0.0:8042 localhost:8042

# DataNode

LocalForward 0.0.0.0:9864 slocalhost:9864

# DataNode

LocalForward 0.0.0.0:9866 slocalhost:9866

# Lesson 13

INGESTION + STORAGE: Azure Synapse Analytics

**Azure HDInsight** [Azure]



Azure HDInsight is a managed analytics service in the cloud. It is basically an implementation of Hadoop in Azure.

HDInsight is a low-cost cloud solution. It includes Apache Hadoop, Spark, Kafka, HBase, Storm, and Interactive Query.

These enable you to run processing tasks over large amounts of data

HDInsight uses a clustered model, like that of Synapse Analytics. HDInsight stores data using Azure Data Lake storage. Hadoop Map/Reduce uses a simple framework to split a task over a large dataset into a series of smaller tasks over subsets of the data that can be run in parallel, and the results then combined.

**To Query Hadoop supports Pig and HiveQL languages. In Spark, data engineers use Spark SQL.**

**Hadoop** includes Apache Hive, HBase, Spark, and Kafka. Hadoop stores data in a file system (HDFS). Spark stores data in memory. This difference in storage makes Spark about 100 times faster.

**HBase** is a NoSQL database built on Hadoop. It's commonly used for search engines. HBase offers automatic failover.

**Storm** is a distributed real-time streamlining analytics solution.

**Kafka** is an open-source platform that's used to compose data pipelines. It offers message queue functionality, which allows users to publish or subscribe to real-time data streams.

[GE] You are a data engineer implementing a lambda architecture on Microsoft Azure. You use an open-source big data solution to collect, process, and maintain data. The analytical data store performs poorly. You must implement a solution that meets the following requirements:

- Provide data warehousing

- Reduce ongoing management activities

- Deliver SQL query responses in less than one second

You need to create an HDInsight cluster to meet the requirements. Which type of cluster should you create? **Apache Spark**

[GE] You need to develop a pipeline for processing data. The pipeline must meet the following requirements:

- Scale up and down resources for cost reduction

- Use an in-memory data processing engine to speed up ETL and machine learning operations.

- Use streaming capabilities

- Provide the ability to code in SQL, Python, Scala, and R

- Integrate workspace collaboration with Git

What should you use? **HDInsight Spark Cluster**

How to create data lake storage:

1

[1] Create hivescript.hql

Logo

Description automatically generated

DROP TABLE IF EXISTS HiveSampleOut;

CREATE EXTERNAL TABLE HiveSampleOut (clientid string, market string, devicemodel string, state string)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ' '

STORED AS TEXTFILE LOCATION 'S{hiveconf:output}';

INSERT OVERWRITE TABLE HiveSampleOut

SELECT

clientid,

market,

devicemodel,

state

FROM hivesampletable

[2] Create Storage Account

*Not gen 2 so no hierarchical storage needed*

hdinsighthive

-- Add Container (blob)

hdinsight

-- Upload hivescript.hql to blob

[6] Create new linked service

Manage > Linked Service > Data Store

**Azure Blob**

[7] Create new linked service

Manage > Linked Service > Compute

**HDInsight**

*Copy your principal ID and key into here*

-- Azure Storage linked service [6]

-- Cluster Type: Hadoop

-- Cluster Size: Hadoop

*Create Hadoop user*

-- OS Type:

ssh username: hduser

ssh password: <passowrd>

[3] Create App Registration (AAD)  
hdinsighthive.com

-- Create client secret

-- Copy App ID

[4] Add Role to Resource Group  
*Add contributor role to hdinsighthive service principal*

Resource Group > IAM > Add Role Assignment

[5] Create new Data Factory

hivefactory

Icon

Description automatically generated

[8] Create data Pipeline

-- Create “HDInsight” Hive activity

-- HDI Cluser: linkedservice1

-- Script: hdinsight/hivescript.hpl

-- Parameters: auto-fill from script

Output: DRIVER\_NAME://CONTAINER\_NAME@STORAGE\_ACCOUNT.blob.core.windows.net/outputfolder

wasbs://hdinsight@hdinsighthive.blob.core.windows.net/outputfolder

Exam Prep:

Storage & HDInsight

1. Get in-depth knowledge of using Azure Data Lake Storage Gen2 for big data requirements. Also, learn about different Hadoop tools discussed in this article.
2. Choose the correct HDInsight Configuration to build open-source analytics solutions. Give special attention to the use cases and get a better understanding of when to use Storm vs Spark etc.

# Lesson 14

STORAGE: Azure Data Lake Storage Gen 2

<https://www.dremio.com/data-lake/azure/>

**Azure Data Lake Storage Gen 2** [Azure]

Icon

Description automatically generatedis a fully managed, elastic, scalable and secure file system that supports HDFS semantics and works with the Apache Hadoop ecosystem. ADLS can store unstructured, semi-structed and structured data. It has no upfront cost (pay-per-use model).

it is a combination of Azure Blob Storage + Azure Data Lake Gen 1.

**Gen 1 Features:**

Azure Data Lake Storage is a Hadoop-compatible data repository that can store any

size or type of data.

**Gen 2 Features:**

- Unlimited scalability

- Cost effective (Pay-per-use model)

- Hadoop compatibility (Manage and access data the same way you would with HDFS)

- Geo-redundant storage

- Zone-redundant storage

- Security support for both access control lists **ACL’s**

- POSIX compliance

- An optimized Azure Blob File System **ABFS** driver (optimized for big data analytics)

Diagram, funnel chart

Description automatically generated

Exam Prep:

Storage

1. Manage the Azure Blob storage lifecycle.
2. Experiment the mentioned example in this post about applying a lifecycle policy. hands-on
3. Learn about Access control in Azure Data Lake Storage Gen2. Pay special
4. attention to Azure AD setup while applying ACLs. hands-on
5. Configure Azure Storage firewalls and virtual networks. hands-on

How to create data lake storage:

1

[1] Create Storage Account

*We use a storage account because ADLS is built on top of blob storage WASB*

adlstorage

-- enable Hierarchical namespace (gen 2)

**Microsoft Azure Storage explorer (Windows/Mac)**[2] Add Container (blob)

images

-- Add new folder

personal

-- Upload file (random local file)

personal

Icon

Description automatically generated

Icon

Description automatically generated

[3] Create new Data Factory

adlsfactory

A picture containing icon

Description automatically generated

[4] Create new SQL Database

adlsdb

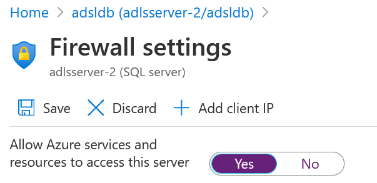
-- Add server

adlsserver

NB: remember to toggle [Azure]

“Allow Azure services and resources to access this server”

-- Add admin user to server



# Lesson 15

PREP & TRAIN: Azure Stream Analytics

[Azure]

<https://docs.microsoft.com/en-us/learn/modules/survey-the-azure-data-platform/8-stream-analytics>

Data engineers use Azure Stream Analytics to process streaming data and respond to data anomalies in real time. You can use Stream Analytics for Internet of Things (IoT) monitoring, web logs, remote patient monitoring, and point of sale (POS) systems.

**Use the declarative Stream Analytics query language to query Azure Stream Analytics**

*Diagram

Description automatically generated*

INGEST*:* configuring data inputs from first-class integration sources. These sources include Azure Event Hubs, Azure IoT Hub, and Azure Blob storage.

Exam Prep:

Azure Stream Analytics

1. Window functions: You must know the practical difference between all the stream analytics windowing functions & their usage (Tumbling, Hopping, Sliding & Session windows). hands-on
2. Learn how to use lookup data in the Azure Stream Analytics in a data streaming pipeline. hands-on
3. Azure Stream Analytics on IoT Edge

# Lesson 16

Data Processing

Lamda Architecture:

<https://sqlwithmanoj.com/2018/02/16/what-is-lambda-architecture-and-what-azure-offers-with-its-new-cosmos-db/>

Lambda architecture is a data-processing architecture designed to handle massive quantities of data by taking advantage of both batch processing and stream processing methods, and minimizing the latency involved in querying big data.

It is a Generic, Scalable, and Fault-tolerant data processing architecture to address batch and speed latency scenarios with big data and map-reduce. It consists of three layers:

Batch Layer: has a master dataset (immutable, append-only set of raw data) and pre-computes the batch views.

Speed Layer: has Batch views for fast queries.

Service Layer:

Diagram

Description automatically generated

# Lesson 17

Monitoring

Exam Prep:

1. Understand the SQL auditing features & do a hand-on experiment on who/when & what got accessed from the Azure SQL DB & WH? hands-on
2. Learn about enabling SQL server automatic tuning & give special attention towards the inheritance with tuning options like Force Plan, Create Index & Drop Index. hands-on
3. Read & understand In-Memory technologies that improve performance without increasing the database service tier. hands-on
4. Understand the materialized view design pattern and think about its uses to boost a slow-performing SQL query. hands-on
5. Learn how to enable and configure logging of diagnostics telemetry for Azure SQL databases. Pay special attention to the metric storage options like Azure SQL Analytics, Azure Event Hubs & Azure Storage. hands-on
6. Imbibe ADLS Gen2 performance optimization techniques. Understand file sizing & data organization into folders.
7. Discover Azure data factory monitoring using Azure monitor and think about use cases like last quarter log analytics and find out different actionable trends. hands-on
8. Learn about on-premises HA data gateway cluster to avoid single points of failure and to load balance traffic across gateways in a cluster.
9. Understand the use of Azure Log Analytics to monitor HDInsight clusters. Pay special attention to “Install HDInsight cluster management solutions”. hands-on

# Lesson 18

CosmosDB

Entities by API

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | CosmosDB | SQL | Cassandra | MongoDB | Gremlin | Table |
| Database |  | Database | Keyspace | Database | Databsae | N/A |
| Container |  | Container | Table | Collection | Graph | Table |
| Item | Azure cosmos item | Document | Row | Document | Node or Edge | Item |

# Misc

[Azure] Data Security

Azure Storage encrypts all data that's written to it. Azure Storage also provides you with fine-grained control over who has access to your data. You'll secure the data by using keys or shared access signatures.

Azure Resource Manager provides a permissions model that uses role-based access control (**RBAC**). Use this functionality to set permissions and assign roles to users, groups, or applications.